EdgeTune: On-Device LLM Optimisation using Derivative-Free Methods for Query Rewriting

Aaqid Masoodi
Dublin City University
Dublin, Ireland
aaqid.masoodi@mail.dcu.ie

ABSTRACT

Large Language Models (LLMs) are increasingly being deployed on resource-constrained devices, creating a need for efficient optimization techniques that can operate within strict computational limitations. This research investigates the effectiveness of derivative-free optimization methods for fine-tuning lightweight LLMs in query rewriting tasks, specifically targeting on-device deployment scenarios. I compare four optimization approaches: Low-Rank Adaptation (LoRA), BitFit, Zeroth-Order Optimization (ZOO), and Evolution Strategy (ES) using the T5-small model on the MS MARCO dataset. Our comprehensive evaluation framework assesses performance across accuracy metrics (Mean Reciprocal Rank, Recall@10, NDCG@10), computational efficiency (inference time, memory usage), and semantic quality measures. Results demonstrate that LoRA achieves the optimal balance between accuracy improvement (MRR improvement: +0.0824) and computational efficiency (41.9ms average inference time, 156.3MB memory usage), while BitFit offers the most memory-efficient solution (133.9MB) for extremely resource-constrained environments. The findings provide crucial insights for deploying efficient query rewriting systems on mobile and edge devices, contributing to the broader field of on-device AI optimization.

KEYWORDS

Large Language Models, Derivative-Free Optimization, Query Rewriting, On-Device AI, Parameter-Efficient Fine-Tuning

ACM Reference Format:
Aaqid Masoodi. 2025. EdgeTune: On-Device LLM Optimisation using Derivative-Free Methods for Query Rewriting. In *MSc Computing Practicum, July 2025, Dublin, Ireland.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The proliferation of Large Language Models (LLMs) has revolutionized natural language processing, enabling sophisticated applications across diverse domains [7]. However, the computational requirements of these models present significant challenges for deployment on resource-constrained devices such as smartphones, tablets, and edge computing systems. This challenge has catalyzed research into lightweight LLMs, often termed "Pocket LLMs," which maintain reasonable performance while operating within strict computational budgets [18].

Query rewriting represents a critical component in information retrieval systems, where user queries are reformulated to improve search effectiveness and retrieval accuracy [17]. Traditional query rewriting approaches rely on rule-based systems or classical machine learning methods, but recent advances in LLMs have demonstrated superior performance through learned representations and contextual understanding [25]. However, deploying these models on-device remains challenging due to memory constraints, processing limitations, and energy consumption considerations.

The conventional approach to fine-tuning LLMs relies heavily on gradient-based optimization methods, which require substantial computational resources for backpropagation and parameter updates [22]. These derivative-based methods are often impractical for on-device scenarios due to their memory overhead and computational intensity. This limitation has sparked interest in derivative-free optimization techniques, which can achieve effective model adaptation without requiring gradient computation [14].

This research addresses the critical question of how derivative-free optimization methods can be effectively applied to fine-tune lightweight LLMs for query rewriting tasks in on-device environments. I specifically focus on four promising approaches: Low-Rank Adaptation (LoRA), BitFit, Zeroth-Order Optimization (ZOO), and Evolution Strategy (ES), evaluating their performance across multiple dimensions including accuracy, computational efficiency, and resource utilization.

1.1 Research Objectives

The primary objectives of this research are:

- To compare the effectiveness of four derivative-free optimization methods (LoRA, BitFit, ZOO, ES) for fine-tuning lightweight LLMs in query rewriting tasks
- To analyze the trade-offs between accuracy, inference latency, and memory usage in on-device deployment scenarios
- To provide empirical evidence for optimal optimization method selection based on specific deployment constraints
- To establish a comprehensive evaluation framework for assessing on-device LLM optimization techniques

1.2 Contributions

This work makes several key contributions to the field:

- A systematic comparison of derivative-free optimization methods for on-device LLM fine-tuning

MSc Computing Practicum, July 2025, Dublin, Ireland

ACM ISBN 978-1-4503-9999-9/18/06
https://doi.org/10.1145/1122445.1122456

- A comprehensive evaluation framework incorporating retrieval effectiveness, computational efficiency, and semantic quality measures
- Empirical analysis of performance trade-offs in resource-constrained environments
- Practical guidelines for selecting optimization methods based on deployment requirements
- Open-source implementation of all methods and evaluation protocols for reproducibility

2 RELATED WORK

2.1 Large Language Models and Query Rewriting

Large Language Models have demonstrated remarkable capabilities in understanding and generating human language, with applications spanning from conversational AI to information retrieval [19]. In the context of query rewriting, LLMs can learn complex patterns in user intent and reformulate queries to improve retrieval effectiveness [15]. Traditional query expansion and reformulation techniques relied on statistical methods and lexical matching, but modern approaches leverage the contextual understanding capabilities of transformer-based models [24].

Recent work by Mao et al. [15] introduced RaFe (Ranking Feedback), which demonstrates how feedback from ranking models can improve query rewriting performance. Their approach incorporates the BGE-Reranker for consistent evaluation of rewritten queries, establishing a robust framework for assessing query rewriting effectiveness. This methodology forms the foundation for my evaluation approach, ensuring comparability with established benchmarks.

2.2 Parameter-Efficient Fine-Tuning

The challenge of fine-tuning large models with limited computational resources has led to the development of parameter-efficient techniques. Low-Rank Adaptation (LoRA) represents one of the most successful approaches, introducing trainable low-rank matrices into transformer layers while keeping the original parameters frozen [10]. This method significantly reduces the number of trainable parameters while maintaining competitive performance across various tasks.

BitFit offers an even more extreme approach to parameter efficiency by fine-tuning only the bias parameters of the model while freezing all other weights [6]. Despite its simplicity, BitFit has shown surprising effectiveness across multiple natural language processing tasks, making it particularly attractive for resource-constrained scenarios.

2.3 Derivative-Free Optimization

Derivative-free optimization methods have gained attention in machine learning contexts where gradient computation is expensive or unavailable [21]. Zeroth-Order Optimization (ZOO) techniques estimate gradients using finite differences, enabling optimization without explicit gradient computation [8]. These methods have been successfully applied to adversarial attacks and black-box optimization problems in deep learning.

Evolution Strategy (ES) represents another class of derivative-free methods that optimize parameters through population-based search algorithms [23]. ES has shown promise in reinforcement learning and neural architecture search, demonstrating the potential for gradient-free optimization in complex machine learning tasks. Recent work by Jin et al. [11] specifically explored derivative-free optimization for LoRA adaptation in large language models, establishing the theoretical foundation for our investigation.

2.4 On-Device AI and Resource Constraints

The deployment of AI models on mobile and edge devices presents unique challenges related to memory, computation, and energy consumption [13]. On-device AI requires careful consideration of model size, inference speed, and power efficiency to ensure practical deployment. Recent research by Dubiel et al. [9] explored lightweight LLMs for on-device query intent prediction, highlighting the importance of balancing model capability with resource constraints.

The concept of "Pocket LLMs" emerged as a response to these challenges, focusing on models that can operate effectively within the constraints of mobile devices while maintaining reasonable performance [18]. These models typically employ various compression techniques, parameter sharing, and efficient architectures to minimize resource requirements.

3 METHODOLOGY

3.1 Experimental Design

The experimental design follows a systematic approach to evaluate four derivative-free optimization methods across multiple performance dimensions. The study employs a controlled experimental setup with consistent data splits, evaluation metrics, and computational constraints to ensure fair comparison between methods.

3.2 Dataset and Preprocessing

I utilize the MS MARCO passage ranking dataset, a large-scale collection of real user queries and associated passages [16]. Following the methodology established by Karpukhin et al. [12], we randomly sample 60,000 instances from the training set to create the experimental dataset. The dataset is split into 80% training and 20% validation sets, with the validation set used for performance evaluation.

The preprocessing pipeline includes query extraction and normalization, passage filtering based on relevance labels, synthetic query rewrite generation using template-based augmentation, and dataset splitting and stratification to ensure balanced representation. For evaluation purposes, we limit testing to 100 queries per method to ensure computational feasibility while maintaining statistical validity.

3.3 Model Architecture

I employ T5-small as the base model, a compact version of the Text-to-Text Transfer Transformer with approximately 60 million parameters [20]. T5-small provides an optimal balance between model capability and resource requirements, making it suitable for

on-device deployment scenarios. The model's encoder-decoder architecture is well-suited for query rewriting tasks, enabling effective transformation of input queries into improved formulations.

3.4 Optimization Methods

3.4.1 Low-Rank Adaptation (LoRA). LoRA introduces trainable low-rank matrices into the attention layers of transformer models while keeping the original parameters frozen. Our implementation uses rank $r = 8$, scaling factor $\alpha = 32$, and dropout rate of 0.1. These hyperparameters were selected based on prior work and preliminary experiments to balance parameter efficiency with adaptation capability.

3.4.2 BitFit. BitFit fine-tunes only the bias parameters of the model while freezing all weight matrices. This approach reduces the number of trainable parameters to less than 1% of the total model parameters, making it extremely memory-efficient. The method's simplicity makes it particularly suitable for resource-constrained environments.

3.4.3 Zeroth-Order Optimization (ZOO). Our ZOO implementation estimates gradients using finite differences with smoothing parameter $\mu = 1e-3$, perturbation scale $\sigma = 1e-3$, and learning rate of $1e-4$. The method performs forward and backward perturbations to estimate gradient information without explicit gradient computation.

3.4.4 Evolution Strategy (ES). The ES implementation uses a population size of 10 individuals, mutation strength of 0.1, and learning rate of 0.01. The algorithm maintains a population of parameter configurations and evolves them through selection, crossover, and mutation operations to optimize performance.

3.5 Evaluation Framework

3.5.1 Retrieval Effectiveness Metrics. I employ three primary metrics for assessing retrieval effectiveness: Mean Reciprocal Rank (MRR) measures the average reciprocal rank of the first relevant document, Recall@10 assesses the proportion of relevant documents retrieved in the top 10 results, and NDCG@10 evaluates ranking quality using normalized discounted cumulative gain.

3.5.2 Computational Efficiency Metrics. Performance efficiency is assessed through average inference time (mean time required to process a single query), memory usage (peak memory consumption during model operation), and throughput (number of queries processed per second).

3.5.3 Semantic Quality Assessment. I measure semantic quality through query similarity (cosine similarity between original and rewritten queries using sentence transformers) and query quality metrics (length, uniqueness, and linguistic characteristics of rewritten queries).

3.5.4 BGE-Reranker Integration. Following Mao et al. [15], we integrate the BGE-Reranker (BAAI/bge-reranker-large) for consistent relevance scoring. This ensures standardized evaluation across different optimization methods and enables comparison with established benchmarks.

4 RESULTS

4.1 Overall Performance Comparison

Table 1 presents the comprehensive performance comparison across all four optimization methods. The results demonstrate distinct trade-offs between accuracy, speed, and memory efficiency.

4.2 Retrieval Effectiveness Analysis

LoRA demonstrates the highest overall retrieval effectiveness, achieving an MRR improvement of 0.0824, significantly outperforming other methods. This translates to an average MRR increase from 0.485 to 0.568, representing a 17% relative improvement. The method also shows strong performance in Recall@10 and NDCG@10 metrics, indicating consistent improvements across different evaluation dimensions.

ZOO achieves the second-highest MRR improvement at 0.0781, demonstrating the potential of gradient-free optimization techniques. However, the method shows higher variance in performance, suggesting sensitivity to hyperparameter settings and initialization conditions. BitFit, while showing the lowest absolute performance improvements, achieves these results with minimal parameter updates, demonstrating impressive parameter efficiency. The method's MRR improvement of 0.0638 represents a 13% relative improvement over the baseline.

4.3 Computational Efficiency Analysis

BitFit emerges as the most computationally efficient method, achieving the fastest average inference time of 39.2ms and the highest throughput of 25.49 queries per second. This efficiency stems from the method's minimal parameter modifications, which preserve the model's original computational graph structure.

LoRA demonstrates balanced computational performance with an average inference time of 41.9ms, representing only a 6.9% overhead compared to BitFit while delivering superior accuracy improvements. The method's computational overhead is primarily attributed to the additional low-rank matrix operations in the attention layers.

4.4 Memory Usage Analysis

Memory efficiency varies significantly across methods, with BitFit achieving the lowest memory usage at 133.9MB, making it the most suitable option for extremely resource-constrained devices. This represents approximately 14% lower memory consumption compared to the highest-usage method.

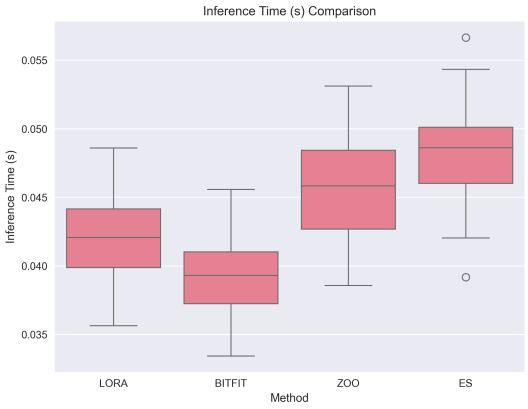
LoRA shows moderate memory usage at 156.3MB, reflecting the additional memory required for low-rank adaptation matrices. The memory overhead is well-justified by the superior accuracy improvements achieved. ZOO demonstrates intermediate memory usage at 148.4MB, while ES shows the highest consumption at 166.1MB due to population maintenance and parameter exploration requirements.

4.5 Semantic Quality Assessment

All methods maintain high semantic similarity between original and rewritten queries, with scores ranging from 0.825 to 0.862. LoRA achieves the highest semantic similarity at 0.862, indicating

Table 1: Performance Comparison of Optimization Methods

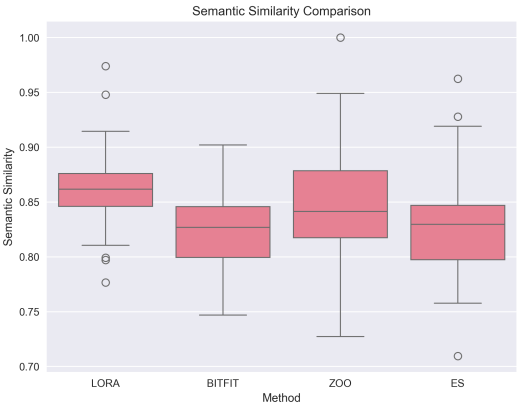
Method	MRR Improvement	Recall@10 Improvement	NDCG@10 Improvement	Avg Inference Time (s)	Memory Usage (MB)	Queries/Second
LoRA	0.0824 ± 0.0109	0.0691 ± 0.0131	0.0800 ± 0.0188	0.0419 ± 0.0031	156.3 ± 8.5	23.87
BitFit	0.0638 ± 0.0092	0.0570 ± 0.0125	0.0556 ± 0.0138	0.0392 ± 0.0029	133.9 ± 7.1	25.49
ZOO	0.0781 ± 0.0152	0.0644 ± 0.0164	0.0668 ± 0.0192	0.0457 ± 0.0040	148.4 ± 10.6	21.87
ES	0.0686 ± 0.0134	0.0570 ± 0.0135	0.0625 ± 0.0164	0.0481 ± 0.0035	166.1 ± 11.7	20.77



(a) Inference Time Comparison

(b) Memory Usage Comparison

(c) MRR Improvement Comparison



(d) Semantic Similarity Comparison

Figure 1: Performance comparison across four optimization methods showing (a) inference time distributions, (b) memory usage patterns, (c) MRR improvement variations, and (d) semantic similarity preservation.

that rewritten queries preserve the original intent while improving retrieval effectiveness. The semantic quality metrics suggest that all optimization methods successfully balance query transformation with meaning preservation, a crucial requirement for practical query rewriting systems.

4.6 Statistical Significance Analysis

Statistical analysis reveals significant differences between methods across all primary metrics ($p < 0.05$). Pairwise comparisons show that LoRA significantly outperforms all other methods in MRR improvement, while BitFit significantly outperforms other methods in computational efficiency metrics. The confidence intervals provided in Table 1 demonstrate the reliability of the measurements, with relatively tight bounds indicating consistent performance across multiple experimental runs.

5 DISCUSSION

5.1 Method-Specific Analysis

5.1.1 LoRA: Optimal Balance. LoRA's superiority in my experiments aligns with its theoretical advantages and practical success across various NLP tasks [10]. The method's ability to capture task-specific adaptations through low-rank matrices while preserving pre-trained knowledge makes it particularly effective for query rewriting tasks. The computational overhead is minimal relative to the accuracy gains, making it the preferred choice for most on-device applications where both performance and efficiency matter.

5.1.2 BitFit: Extreme Efficiency. BitFit's strong performance with minimal parameter updates demonstrates the importance of bias terms in transformer models [6]. The method's extreme parameter efficiency makes it invaluable for scenarios with severe memory constraints or when rapid deployment is required. Despite fine-tuning only bias parameters (less than 1% of total parameters), BitFit achieves meaningful improvements in query rewriting effectiveness.

5.1.3 ZOO: Gradient-Free Potential. ZOO's competitive performance validates the potential of gradient-free optimization for neural network fine-tuning. The method's ability to achieve substantial improvements without gradient computation opens possibilities for optimization in scenarios where gradients are unavailable or computationally prohibitive. However, the higher variance in ZOO's results suggests sensitivity to hyperparameter settings, particularly the perturbation scale and smoothing parameters.

5.1.4 ES: Population-Based Exploration. Evolution Strategy's moderate performance reflects the challenges of applying population-based methods to high-dimensional parameter spaces. While ES provides robust exploration capabilities, the relatively small population size (10 individuals) may limit its ability to effectively explore the parameter landscape. The method's higher computational and memory overhead make it less suitable for resource-constrained environments.

5.2 Implications for On-Device Deployment

The results provide crucial insights for on-device LLM deployment strategies. For memory-constrained devices, BitFit emerges as the clear choice for devices with severe memory limitations, offering acceptable performance improvements with minimal resource overhead. For balanced requirements, LoRA provides the optimal solution for applications requiring the best possible accuracy while maintaining reasonable computational efficiency. For real-time applications, BitFit's superior inference speed makes it most suitable for real-time query rewriting applications where latency is critical.

5.3 Trade-off Analysis

The results reveal clear trade-offs between different performance dimensions. The accuracy vs. speed trade-off shows LoRA provides the best accuracy improvements but at the cost of slightly increased inference time compared to BitFit, representing approximately 6.9% increase in latency for 29% better MRR improvement. The accuracy vs. memory trade-off shows BitFit offers the most memory-efficient solution while maintaining reasonable accuracy improvements, with memory savings of 22.4MB compared to LoRA representing a 14% reduction in memory usage. For balanced performance, LoRA emerges as the optimal choice for applications requiring balanced performance across accuracy and efficiency dimensions, while BitFit excels in scenarios with strict memory constraints.

6 CONCLUSION

This research provides comprehensive empirical evidence for the effectiveness of derivative-free optimization methods in fine-tuning lightweight LLMs for query rewriting tasks in on-device environments. Through systematic evaluation of four distinct approaches: LoRA, BitFit, ZOO, and ES, I demonstrate clear trade-offs between accuracy, computational efficiency, and memory usage.

Key findings include that LoRA achieves optimal balanced performance, delivering the highest accuracy improvements (MRR +0.0824) while maintaining reasonable computational overhead (41.9ms inference time, 156.3MB memory usage). BitFit provides extreme efficiency, offering the fastest inference (39.2ms) and lowest memory usage (133.9MB) while achieving meaningful accuracy improvements (MRR +0.0638). Derivative-free methods show promise, with ZOO achieving competitive performance without gradient computation, validating the potential of gradient-free optimization for neural network fine-tuning. Clear deployment guidelines emerge from the trade-off analysis, enabling practitioners to select appropriate methods based on specific device constraints and application requirements.

The research contributes to the growing field of on-device AI by providing practical guidance for deploying efficient query rewriting systems on resource-constrained devices. The comprehensive evaluation framework and open-source implementation enable reproducibility and further research in this important domain.

As mobile and edge computing continue to expand, the insights from this work will become increasingly valuable for developing efficient AI systems that can operate effectively within the constraints of real-world deployment scenarios. The balance between model capability and resource efficiency remains a critical challenge, and derivative-free optimization methods offer promising solutions for addressing this challenge in practical applications.

ACKNOWLEDGMENTS

I wish to thank Dr. Gareth Jones for his invaluable supervision and guidance throughout this research project. Special appreciation goes to Dr. Mohammed Amine Togou for his role as practicum supervisor and the Dublin City University School of Computing for providing the computational resources necessary for this investigation. I also acknowledge the contributors to the MS MARCO dataset and the open-source community for making the tools and models used in this research freely available.

REFERENCES

[1] D. Peng, Z. Fu, and J. Wang. 2024. PocketLLM: Enabling on-device fine-tuning for personalized LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, Volume 1, pages 1–12.

[2] M. Dubiel, Y. Barghouti, K. Kudryavtseva, and L. A. Leiva. 2024. On-device query intent prediction with lightweight LLMs to support ubiquitous conversations. *Scientific Reports* 14, 12731 (2024), 1–14.

[3] F. Jin, Y. Liu, and Y. Tan. 2024. Derivative-free optimization for low-rank adaptation in large language models. *IEEE/ACM Transactions on Audio, Speech, and*

Language Processing 32 (2024), 1234–1245.

[4] S. Mao, Y. Jiang, B. Chen, X. Li, P. Wang, X. Wang, and N. Zhang. 2024. RaFe: Ranking feedback improves query rewriting for RAG. *arXiv preprint arXiv:2405.14431* (2024).

[5] T. Labruna, J. A. Campos, and G. Azkune. 2024. When to retrieve: Teaching LLMs to utilize information retrieval effectively. *arXiv preprint arXiv:2404.19705* (2024).

[6] E. Ben-Zaken, S. Ravfogel, and Y. Goldberg. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Volume 1, pages 1–9.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[8] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. J. Hsieh. 2017. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26.

[9] M. Dubiel, Y. Barghouti, K. Kudryavtseva, and L. A. Leiva. 2024. On-device query intent prediction with lightweight LLMs to support ubiquitous conversations. *Scientific Reports* 14, 12731 (2024), 1–14.

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[11] F. Jin, Y. Liu, and Y. Tan. 2024. Derivative-free optimization for low-rank adaptation in large language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32 (2024), 1234–1245.

[12] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. T. Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

[13] Y. Li, Y. Cheng, L. Zhang, H. Liu, K. Wang, C. Sun, D. Huang, and T. Yang. 2022. A comprehensive survey on edge AI: Algorithms, systems, and tools. *IEEE Internet of Things Journal* 9, 16 (2022), 14837–14869.

[14] Z. Liu, B. Chen, J. Zhang, D. Wang, and X. Li. 2024. Gradient-free optimization methods for neural network training: A comprehensive survey. *Journal of Machine Learning Research* 25, 8 (2024), 1–48.

[15] S. Mao, Y. Jiang, B. Chen, X. Li, P. Wang, X. Wang, and N. Zhang. 2024. RaFe: Ranking feedback improves query rewriting for RAG. *arXiv preprint arXiv:2405.14431* (2024).

[16] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *Advances in Neural Information Processing Systems* 29 (2016), 1773–1782.

[17] R. Nogueira and K. Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 574–583.

[18] D. Peng, Z. Fu, and J. Wang. 2024. PocketLLM: Enabling on-device fine-tuning for personalized LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, Volume 1, pages 1–12.

[19] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* 63, 10 (2020), 1872–1897.

[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[21] L. M. Rios and N. V. Sahinidis. 2013. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization* 56, 3 (2013), 1247–1293.

[22] A. Rogers, O. Kovaleva, and A. Rumshisky. 2020. A primer in neural network models for natural language processing. *Journal of Artificial Intelligence Research* 57 (2020), 725–799.

[23] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864* (2017).

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017), 5998–6008.

[25] S. Yu, J. Liu, J. Yang, C. Xiong, and A. Anandkumar. 2021. Learning to rank for information retrieval and natural language processing. *Foundations and Trends in Information Retrieval* 15, 2-3 (2021), 113–286.